·····CPU·····

·····CPU·NUMA·····BAM·····

·····CPU ·····NUMA ·····
····· CPU ·····

·····

numactl --hardware · lscpu ·····/proc/self/status ·· Cpus_allowed_list · Mems_allowed_list ·····
····· cpuset ····· Python ····· 0-3,8 ····· CPU ····
·····

··· | ····· | ··

numactl --hardware | ·····CPU····· | ·····

Cpus_allowed_list | ·····CPU | ·····CPU···

Mems_allowed_list | ····· | ·····

numastat -p PID | ····· | ·····

```
from pathlib import Path
import sys

def expand(text):
    result = set()
    for part in text.strip().split(','):
        bounds = part.split('-')
        if len(bounds) not in (1, 2) or not all(x.isdigit() for x in bounds):
            raise ValueError('····· 0-3,8')
        lo, hi = int(bounds[0]), int(bounds[-1])
        if lo > hi:
            raise ValueError('·····')
        result.update(range(lo, hi + 1))
    if not result:
        raise ValueError('·····')
    return result

status = {}
for line in Path('/proc/self/status').read_text().splitlines():
    if ':' in line:
        key, value = line.split(':', 1)
        status[key] = value.strip()
cpus, nodes = expand(sys.argv[1]), expand(sys.argv[2])
if not cpus.issubset(expand(status['Cpus_allowed_list'])):
    raise SystemExit('CPU·····')
if not nodes.issubset(expand(status['Mems_allowed_list'])):
    raise SystemExit('·····')
print('Allowed subset verified')
```

·····

··· samtools sort····· NUMA ····· CPUSET·NODESE
T·····BAM ·····

```
set -euo pipefail
: "${CPUSET:?··CPU·· 0-3}" "${NODESET:?····· 0,1}"
: "${BAM:?·····BAM}" "${WORK:?·····}"
: "${THREADS:?···}" "${MEM:?····· 256M}"
[[ "$THREADS" =~ ^[1-9][0-9]*$ ]] || exit 2
test -s "$BAM" && test -d "$WORK"
python check_numa.py "$CPUSET" "$NODESET"
root=$(mktemp -d "$WORK/numa-sort.XXXXXX")
numactl --version > "$root/numactl-version.txt" 2>&1
numactl --hardware > "$root/hardware.txt"
grep -E '^(Cpus_allowed_list|Mems_allowed_list):' /proc/self/status > "$root/allowed.txt"
samtools quickcheck -v "$BAM"
index=0
for mode in local interleave interleave local; do
  index=$((index+1)); d="$root/$index-$mode"; mkdir "$d"
  if [[ "$mode" = local ]]; then policy=(--localalloc)
  else policy=(--interleave="$NODESET"); fi
  tmp=$(mktemp -d "$WORK/numa-tmp.XXXXXX")
  /usr/bin/time -v -o "$d/time.txt"     numactl --physcpubind="$CPUSET" "${policy[@]}"     samtools sort -@ "$THREADS" -m "$MEM" -T "$tmp/prefix"     -o "$d/sort
ed.bam" "$BAM" > "$d/stdout.txt" 2> "$d/stderr.txt"
  samtools quickcheck -v "$d/sorted.bam"
  samtools view -c "$d/sorted.bam" > "$d/record-count.txt"
done
printf 'Results: %s
' "$root"
```

numactl --localalloc ·····--interleave ····· cpuset ·
···samtools · -m ····· -m·····

·····

……samtools……
……CPU……
……BAM……

……

…… RSS……quickcheck ····· BAM ……
…… samtools ···· numastat -p PID…… shell · PID ……
……

…… peak RSS……CPU……CPU……NUMA……
……I/O……
……

……
…… CPU……

……samtools …BAM ……
……CPU/……

····

Linux··NUMA·····cpuset·· https://docs.kernel.org/admin-guide/mm/numa_memory_policy.html

numactl……CPU·· https://github.com/numactl/numactl/blob/master/numactl.8

samtools sort…… https://www.htslib.org/doc/samtools-sort.html